

9 Architecture Viewpoints

9.1 Introduction

Establishing and maintaining a coherent enterprise architecture is clearly a complex task, because it involves many different people with differing backgrounds using various notations. In order to get a handle on this complexity, researchers have initially focused on the definition of architectural frameworks for classifying and positioning the various architectural descriptions with respect to each other (e.g., the Zachman framework [5], [9]). A problem with looking at enterprise architecture through the lens of an architectural framework is that it categorizes and divides architectural descriptions rather than providing insight into their coherence.

ArchiMate advocates a more flexible approach in which architects and other stakeholders can define their own views on the enterprise architecture. In this approach, views are specified by *viewpoints*. Viewpoints define abstractions on the set of models representing the enterprise architecture, each aimed at a particular type of stakeholder and addressing a particular set of concerns. Viewpoints can both be used to view certain aspects in isolation, and for relating two or more aspects.

The notion of viewpoint-oriented architecture has been around for a while in requirements and software engineering. In the 1990s, a substantial number of researchers worked on what was phrased as “the multiple perspectives problem” [16], [17]. By this term they referred to the problem of how to organize and guide (software) development in a setting with many actors, using diverse representation schemes, having diverse domain knowledge and different development strategies. A general framework has been developed in order to address the diverse issues related to this problem [16], [17]. In this framework, a viewpoint combines the notion of “actor”, “role”, or “agent” in the development process with the idea of a “perspective” or “view” which an actor maintains. More precisely, viewpoints are defined as loosely coupled, locally managed, distributable objects; thus containing identity, state, and behavior. A viewpoint is more than a “partial specification”; in addition, it contains partial knowledge of how to develop that partial specification. These early ideas on viewpoint-oriented software engineering have found their way into ISO/IEC 42010:2007 [2] on which we have based our definitions below.

As a result of these ideas, several architecture frameworks can be found in the field of literature, which are essentially viewpoint classification schemes. For example, the Zachman framework [5], [9] divides the enterprise architecture into 36 different enterprise-wide “architectures” (i.e., viewpoints). Tapscott and Caston’s framework [18] distinguishes five different and complementing viewpoints: business, work, information, application, and technology. Kruchten [19] introduces the “4+1” method, in which four views (logic, process, development, and physical), each having its own notation, are coupled through a fifth view: the scenario view illustrating the collaboration between the other four views.

Viewpoints are also prominently present in the ISO standardized Reference Model for Open Distributed Processing (RM-ODP) [7]. The RM-ODP identifies five viewpoints from which to specify ODP systems, each focusing on a particular area of concern; i.e., enterprise, information, computational, engineering, and technology. It is claimed that the ODP viewpoints form a necessary and sufficient set to meet the needs of ODP standards. More recently, the term “viewpoint” is also used in OMG’s Model Driven Architecture (MDA) initiative to refer to the different model types; i.e., Platform-Independent Model (PIM) and Platform-Specific Model (PSM) [20]. Hence, we conclude that the use of viewpoints and architectural views are well-established concepts in software architecture.

In the domain of enterprise architecture, the TOGAF framework describes a taxonomy of views for different categories of stakeholders. Next to this description of views, TOGAF also provides guidelines for the development and use of viewpoints and views in enterprise architecture models.

The views and viewpoints proposed by any of the above mentioned frameworks should not be considered in isolation: views are inter-related and, often, it is exactly a combination of views together with their underlying inter-dependency relationships the best way to describe and communicate a piece of architecture. It should, however, be noted that

views and viewpoints have a limiting character. They are eventually a restriction of the whole system (and architecture) to a partial number of aspects – a view is just a partial incomplete depiction of the system.

9.2 Views, Viewpoints, and Stakeholders

Views are an ideal mechanism to purposefully convey information about architecture areas. In general, a *view* is defined as a part of an architecture description that addresses a set of related concerns and is addressed to a set of stakeholders. A view is specified by means of a *viewpoint*, which prescribes the concepts, models, analysis techniques, and visualizations that are provided by the view. Simply put, a view is what you see, and a viewpoint is where you are looking from.

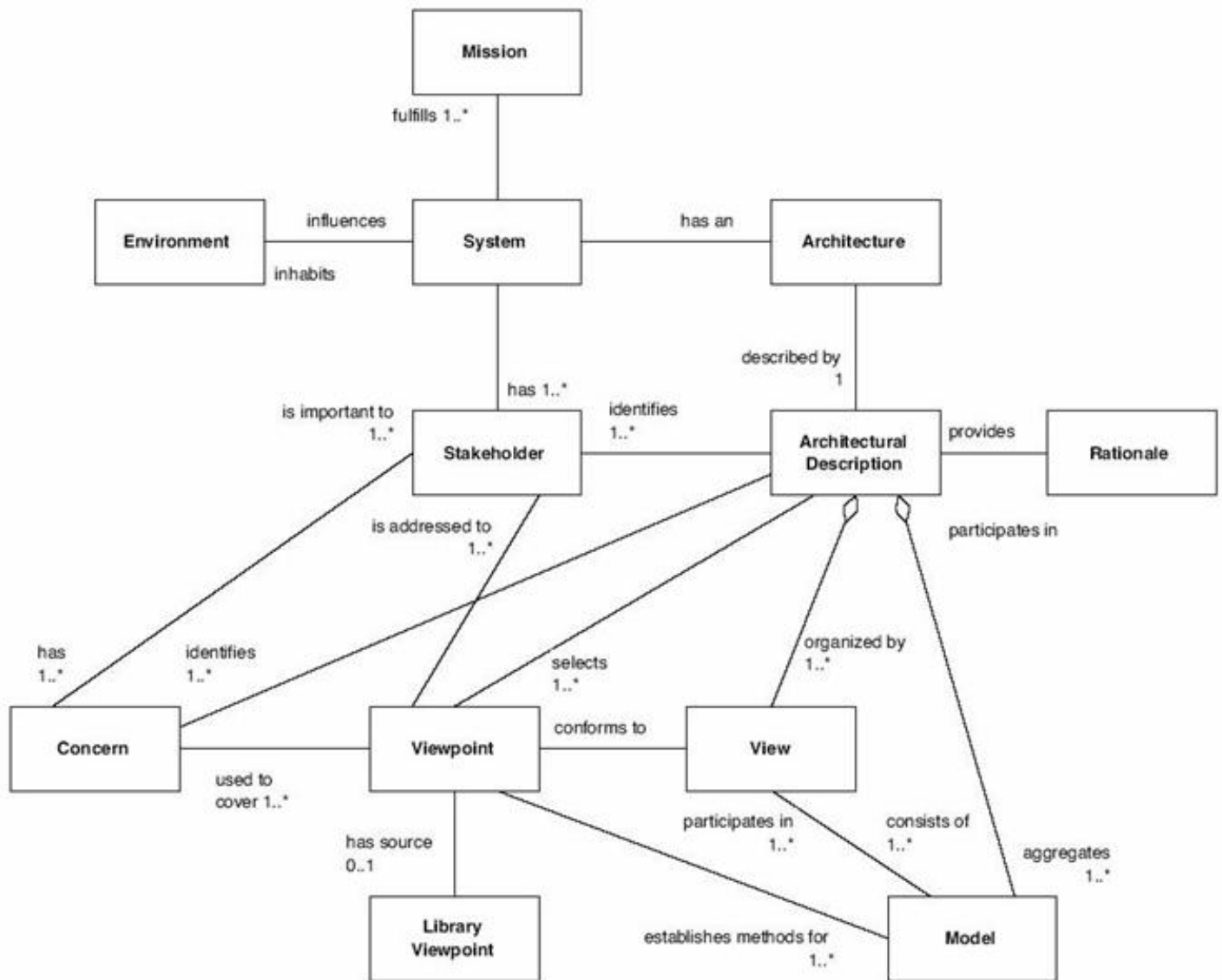


Figure 53: Conceptual Model of Architectural Description (from [2])

Viewpoints are a means to focus on particular aspects of the architecture. These aspects are determined by the concerns of a stakeholder with whom communication takes place. What should and should not be visible from a specific viewpoint is therefore entirely dependent on the argumentation with respect to a stakeholder's concerns.

Viewpoints are designed for the purpose of communicating certain aspects of an architecture. The communication enabled by a viewpoint can be strictly informative, but in general will be bi-directional. The architect informs stakeholders, and stakeholders give their feedback (critique or consent) on the presented aspects. What is and what is not shown in a view depends on the scope of the viewpoint and on what is relevant to the concerns of the stakeholder.

Ideally, these are the same; i.e., the viewpoint is designed with specific concerns of a stakeholder in mind. Relevance to a stakeholder's concern, therefore, is *the* selection criterion that is used to determine which objects and relations are to appear in a view.

The following are examples of stakeholders and concerns as a basis for the specification of viewpoints:

- *End user*: For example, what are the consequences for his work and workplace?
- *Architect*: What is the consequence for the maintainability of a system, with respect to corrective, preventive, and adaptive maintenance?
- *Upper-level management*: How can we ensure our policies are followed in the development and operation of processes and systems? What is the impact of decisions (on personnel, finance, ICT, etc.)?
- *Operational manager*, responsible for exploitation or maintenance: For example, what new technologies are there to prepare for? Is there a need to adapt maintenance processes? What is the impact of changes to existing applications? How secure are my systems?
- *Project manager*, responsible for the development of new applications: What are the relevant domains and their relations? What is the dependence of business processes on the applications to be built? What is their expected performance?
- *Developer*: What are the modifications with respect to the current situation that need to be done?

9.3 Viewpoint Classification

An architect is confronted with many different types of stakeholders and concerns. To help him in selecting the right viewpoints for the task at hand, we introduce a framework for the definition and classification of viewpoints and views. The framework is based on two dimensions: *purpose* and *content*. The following three types of architecture support the purpose dimension of architecture views:

- *Designing*: Design viewpoints support architects and designers in the design process from initial sketch to detailed design. Typically, design viewpoints consist of diagrams, like those used in, for example, UML.
- *Deciding*: Decision support viewpoints assist managers in the process of decision-making by offering insight into cross-domain architecture relations, typically through projections and intersections of underlying models, but also by means of analytical techniques. Typical examples are cross-reference tables, landscape maps, lists, and reports.
- *Informing*: Informing viewpoints help to inform any stakeholder about the enterprise architecture, in order to achieve understanding, obtain commitment, and convince adversaries. Typical examples are illustrations, animations, cartoons, flyers, etc.

The goal of this classification is to assist architects and others find suitable viewpoints given their task at hand; i.e., the purpose that a view must serve and the content it should display. With the help of this framework, it is easier to find typical viewpoints that might be useful in a given situation. This implies that we do not provide an orthogonal categorization of each viewpoint into one of three classes; these categories are not exclusive in the sense that a viewpoint in one category cannot be applied to achieve another type of support. For instance, some decision support viewpoints may be used to communicate to any other stakeholders as well.

For characterizing the content of a view we define the following abstraction levels:

- *Details*: Views on the detailed level typically consider one layer and one aspect from the ArchiMate framework. Typical stakeholders are a software engineer responsible for design and implementation of a software component or a process owner responsible for effective and efficient process execution. Examples of views are a BPMN process diagram and a UML class diagram.

- **Coherence:** At the coherence abstraction level, multiple layers or multiple aspects are spanned. Extending the view to more than one layer or aspect enables the stakeholder to focus on architecture relations like process-uses-system (multiple layer) or application-uses-object (multiple aspect). Typical stakeholders are operational managers responsible for a collection of IT services or business processes.
- **Overview:** The overview abstraction level addresses both multiple layers and multiple aspects. Typically, such overviews are addressed to enterprise architects and decision-makers, such as CEOs and CIOs.

In Figure 54, the dimensions of purpose and abstraction level are visualized in a single picture, together with examples of stakeholders. Table 5 and Table 6 summarize the different purposes and abstraction levels.

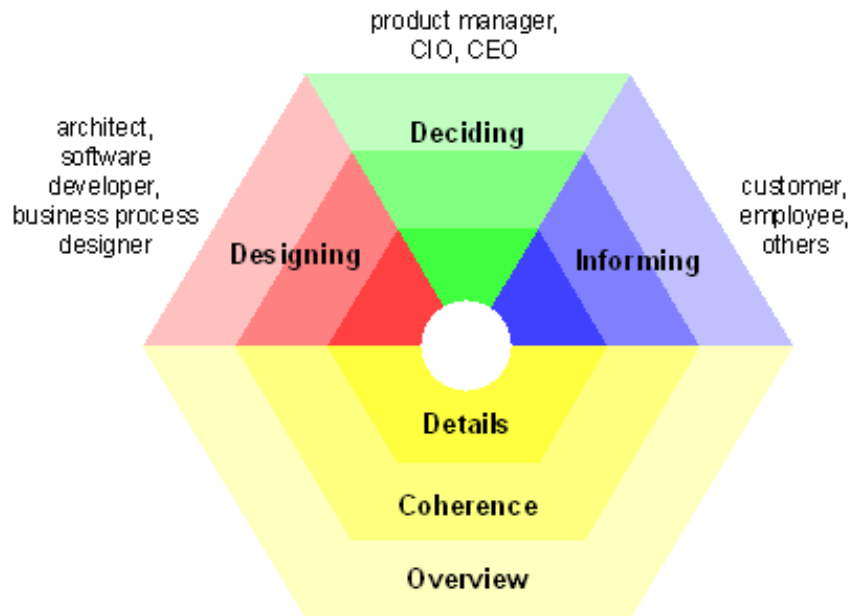


Figure 54: Classification of Enterprise Architecture Viewpoints

Table 5: Viewpoint Purpose

	Typical Stakeholders	Purpose	Examples
Designing	architect, software developer, business process designer	navigate, design, support design decisions, compare alternatives	UML diagram, BPMN diagram, flowchart, ER diagram
Deciding	manager, CIO, CEO	decision-making	cross-reference table, landscape map, list, report
Informing	employee, customer, others	explain, convince, obtain commitment	animation, cartoon, process illustration, chart

Table 6: Viewpoint Abstraction Levels

	Typical Stakeholders	Purpose	Examples
Details	software engineer, process owner	design, manage	UML class diagram, BPMN process diagram
Coherence	operational managers	analyze dependencies, impact of-change	views expressing relations like “use”, “realize”, and “assign”
Overview	enterprise architect, CIO, CEO	change management	landscape map

9.4 Basic Viewpoints in ArchiMate

The basic viewpoint in ArchiMate is a selection of a relevant subset of the ArchiMate concepts (and their relations) and the representation of that part of an architecture that is expressed in different diagrams. A set of such viewpoints was developed based on practical experience. Some of these viewpoints have a scope that is limited to a single layer or aspect. Thus, the Business Function and Business Process viewpoints show the two main perspectives on the business behavior; the Organization viewpoint depicts the structure of the enterprise in terms of its departments, roles, etc.; the Information Structure viewpoint describes the information and data used; the Application Structure, Behavior, and Co-operation viewpoints contain the applications and components and their mutual relations; and the Infrastructure viewpoint shows the infrastructure and platforms underlying the enterprise's information systems in terms of networks, devices, and system software. Other viewpoints link multiple layers and/or aspects: the Actor Co-operation and Product viewpoints relate the enterprise to its environment; the Application Usage viewpoint relates applications to their use in, for example, business processes; and the Deployment viewpoint shows how applications are mapped onto the underlying infrastructure.

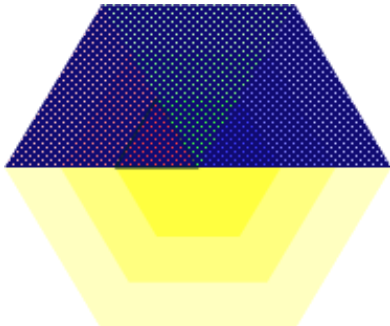
In the following sections, all ArchiMate viewpoints are separately described in detail. For each viewpoint the comprised concepts and relations, the guidelines for the viewpoint use, and the goal and target group and of the viewpoint are indicated. Furthermore, each viewpoint description contains example models. For more details on the goal and use of viewpoints, refer to [3], Chapter 7.

9.4.1 Introductory Viewpoint

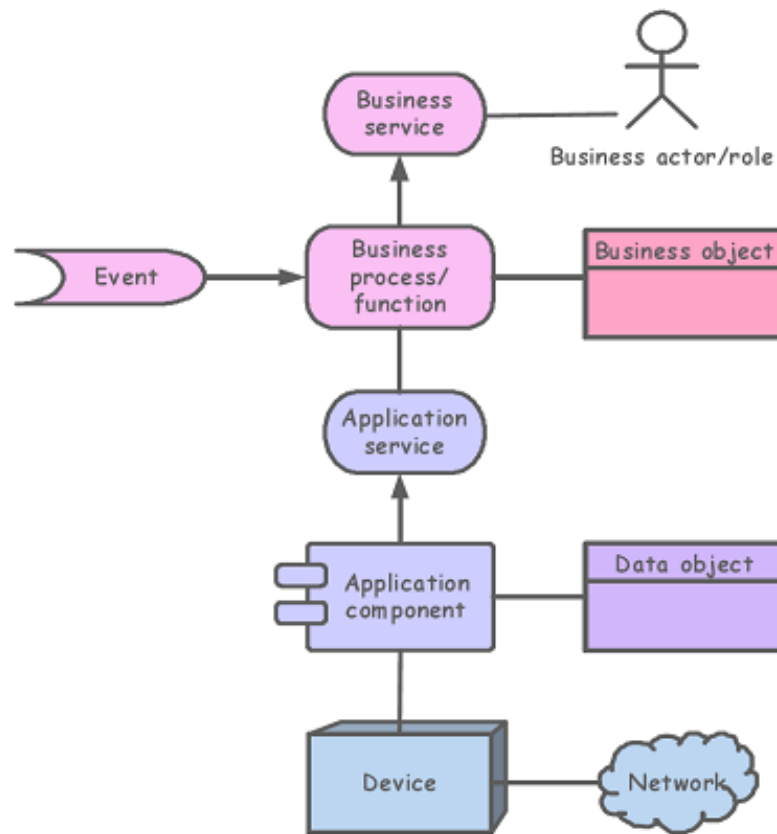
The Introductory viewpoint forms a subset of the full ArchiMate language using a simplified notation. It is typically used at the start of a design trajectory, when not everything needs to be detailed yet, or to explain the essence of an architecture model to non-architects that require a simpler notation. Another use of this basic, less formal viewpoint is that it tries to avoid the impression that the architectural design is already fixed, an idea that may easily arise when using a more formal, highly structured or detailed visualization.

We use a simplified notation for the concepts, and for the relations. All relations except “triggering” and “realization” are denoted by simple lines; “realization” has an arrow in the direction of the realized service; “triggering” is also represented by an arrow. The concepts are denoted with slightly thicker lines and rounded corners, which give a less formal impression. The example below illustrates this notation. On purpose, the layout of this example is not as “straight” as an ordinary architecture diagram; this serves to avoid the idea that the design is already fixed.

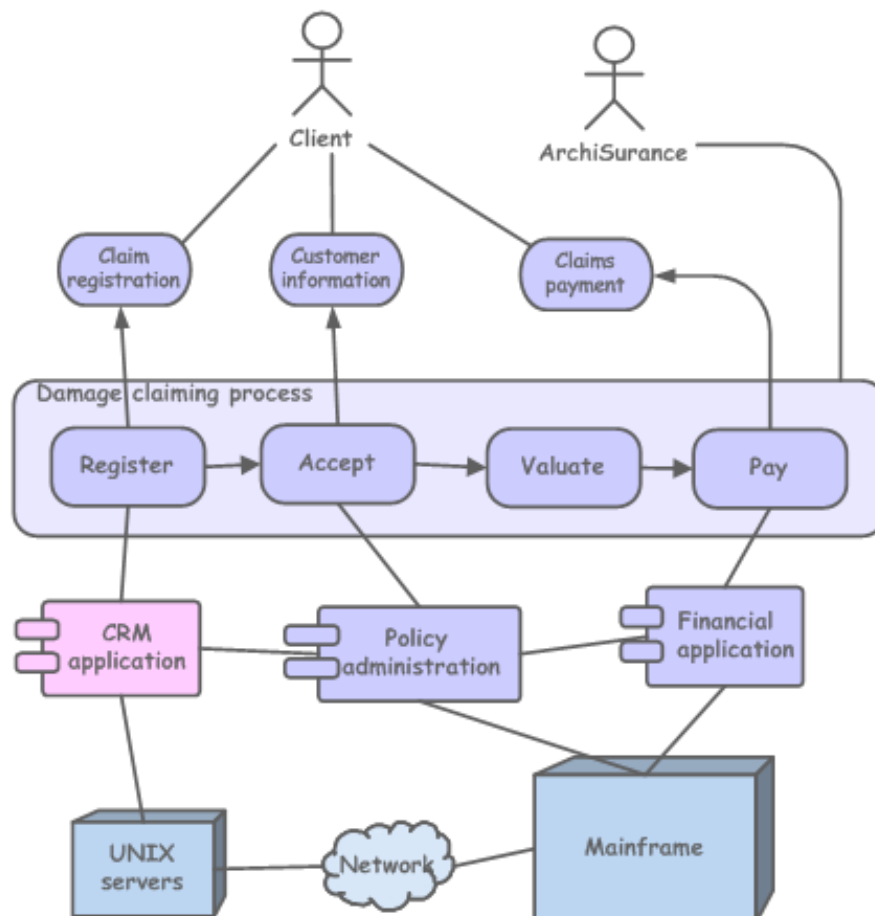
Table 7: Introductory Viewpoint Description

Introductory Viewpoint		
Stakeholders	Enterprise architects, managers	
Concerns	Make design choices visible, convince stakeholders	
Purpose	Designing, deciding, informing	
Abstraction Level	Coherence, Overview, Detail	
Layer	Business, Application, and Technology layers (see also Figure 5)	
Aspects	Structure, behavior, information (see also Figure 5)	

Concepts and Relations



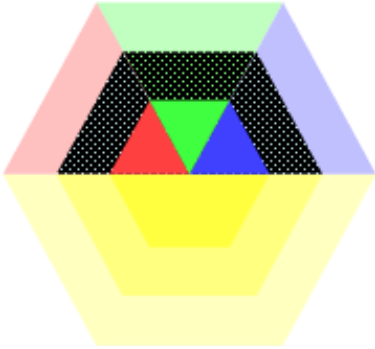
Example



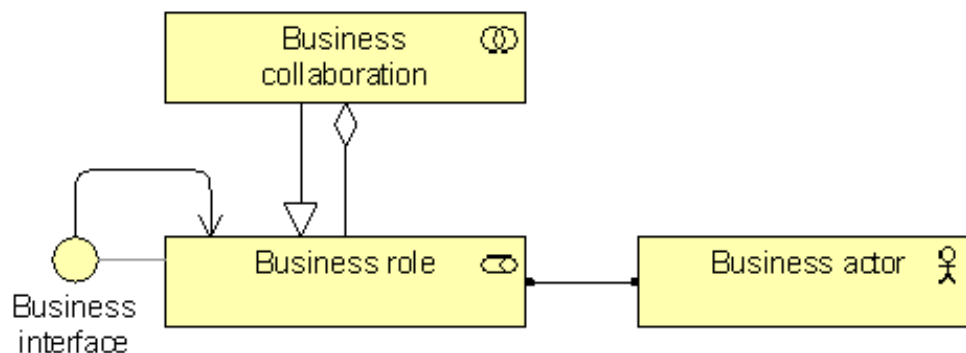
9.4.2 Organization Viewpoint

The Organization viewpoint focuses on the (internal) organization of a company, a department, a network of companies, or of another organizational entity. It is possible to present models in this viewpoint as nested block diagrams, but also in a more traditional way, such as organizational charts. The Organization viewpoint is very useful in identifying competencies, authority, and responsibilities in an organization.

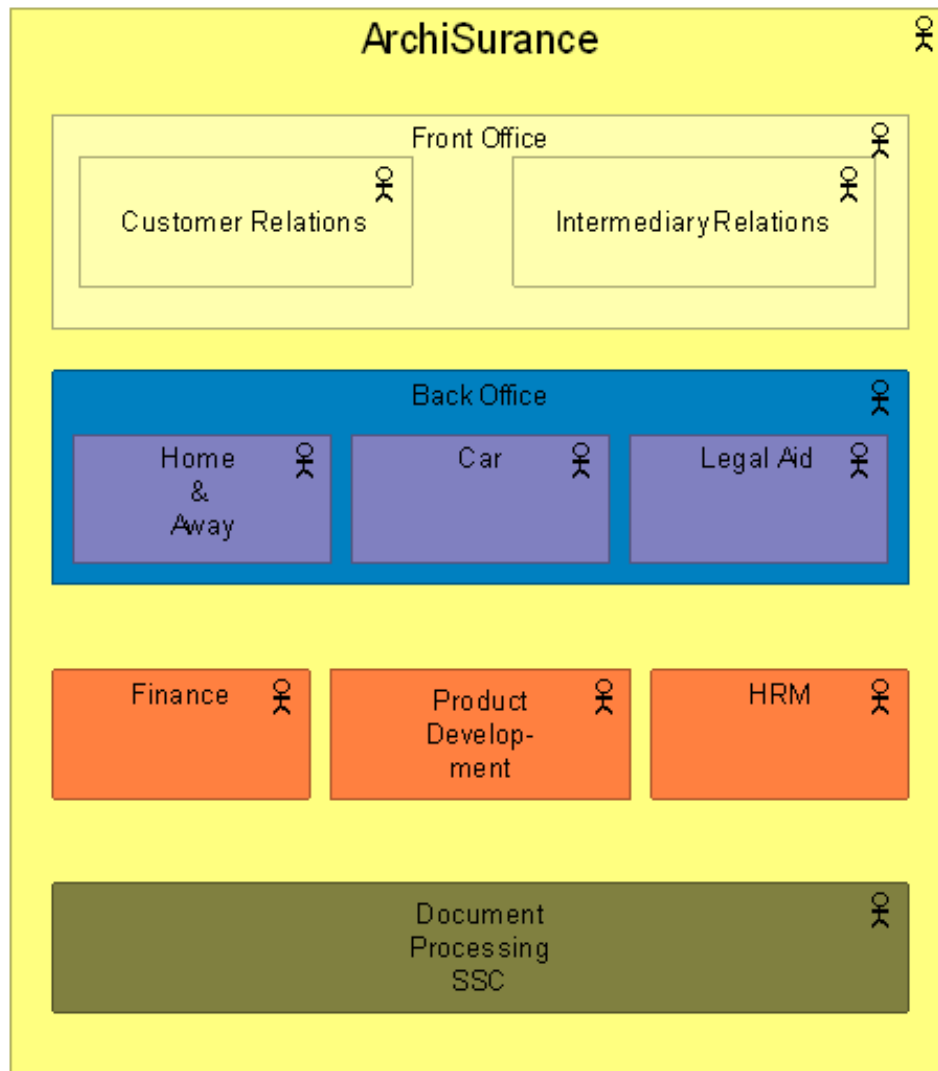
Table 8: Organization Viewpoint Description

Organization Viewpoint			
Stakeholders	Enterprise, process and domain architects, managers, employees, shareholders		
Concerns	Identification of competencies, authority, and responsibilities		
Purpose	Designing, deciding, informing		
Abstraction Level	Coherence		
Layer	Business layer (see also Figure 5)		
Aspects	Structure (see also Figure 5)		

Concepts and Relations



Example




9.4.3 Actor Co-operation Viewpoint

The Actor Co-operation viewpoint focuses on the relations of actors with each other and their environment. A common example of this is the “context diagram”, which puts an organization into its environment, consisting of external parties such as customers, suppliers, and other business partners. It is very useful in determining external dependencies and collaborations and shows the value chain or network in which the actor operates.

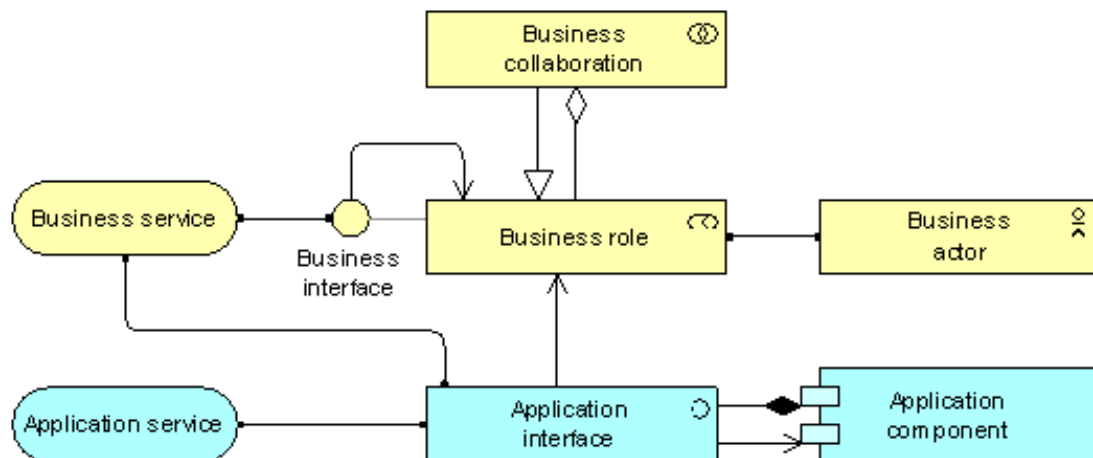
Another important use of the Actor Co-operation viewpoint is in showing how a number of co-operating business actors and/or application components together realize a business process. Hence, in this view, both business actors or roles and application components may occur.

Table 9: Actor Co-operation Viewpoint Description

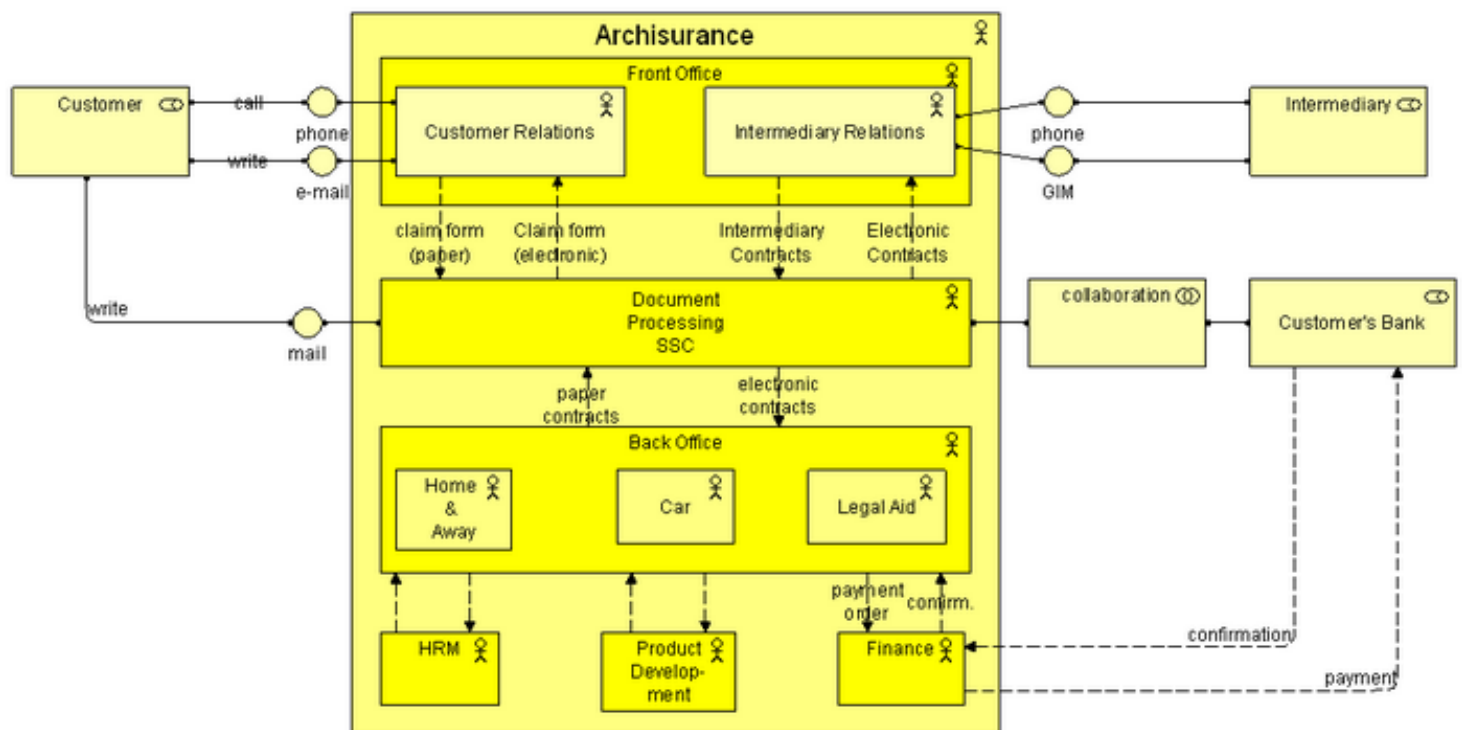
Actor Co-operation Viewpoint		
Stakeholders	Enterprise, process, and domain architects	
Concerns	Relations of actors with their environment	
Purpose	Designing, deciding, informing	

		
Abstraction Level	Detail	
Layer	Business layer (application layer) (see also Figure 5)	
Aspects	Structure, behavior (see also Figure 5)	

Concepts and Relations




Example



9.4.4 Business Function Viewpoint

The Business Function viewpoint shows the main business functions of an organization and their relations in terms of the flows of information, value, or goods between them. Business functions are used to represent the most stable aspects of a company in terms of the primary activities it performs, regardless of organizational changes or technological developments. Therefore, the business function architecture of companies that operate in the same market often exhibit close similarities. The business function viewpoint thus provides high-level insight in the general operations of the company, and can be used to identify necessary competencies, or to structure an organization according to its main activities.

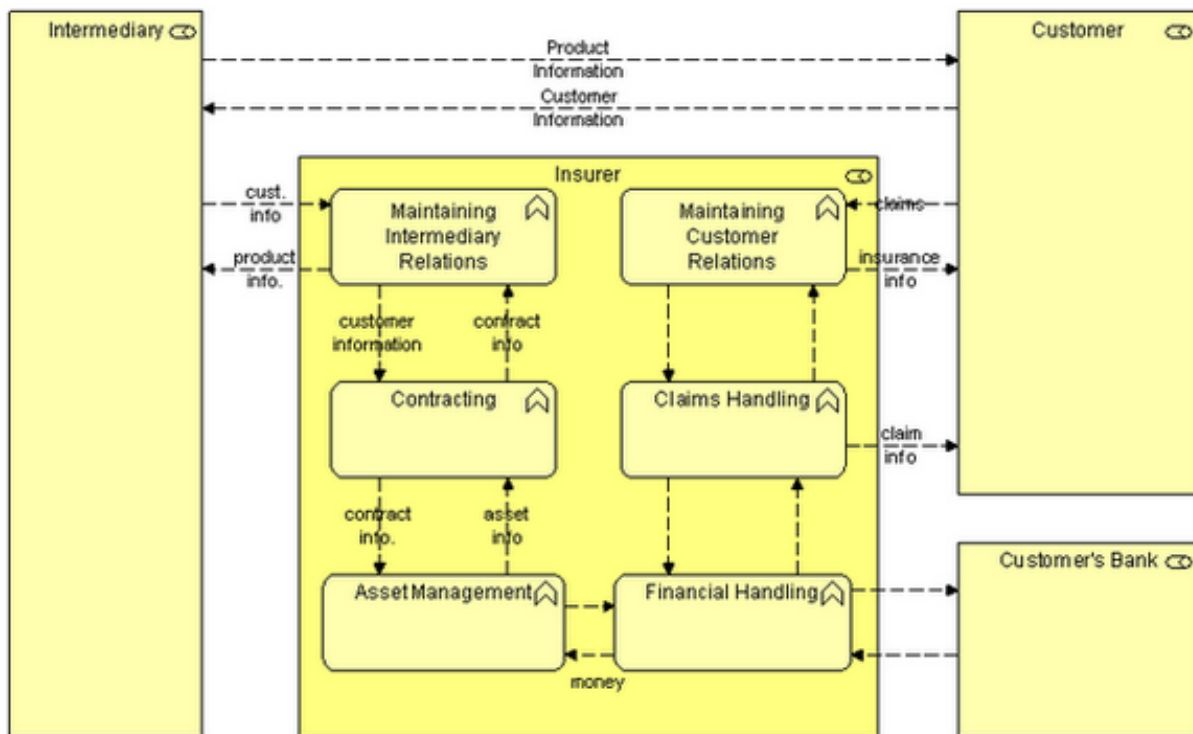
Table 10: Business Function Viewpoint Description

Business Function Viewpoint		
Stakeholders	Enterprise, process, and domain architects	
Concerns	Identification of competencies, identification of main activities, reduction of complexity	
Purpose	Designing	
Abstraction Level	Coherence	
Layer	Business layer (see also Figure 5)	
Aspects	Behavior, structure (see also Figure 5)	

Concepts and Relations



Example




9.4.5 Business Process Viewpoint

The Business Process viewpoint is used to show the high-level structure and composition of one or more business processes. Next to the processes themselves, this viewpoint contains other directly related concepts, such as:

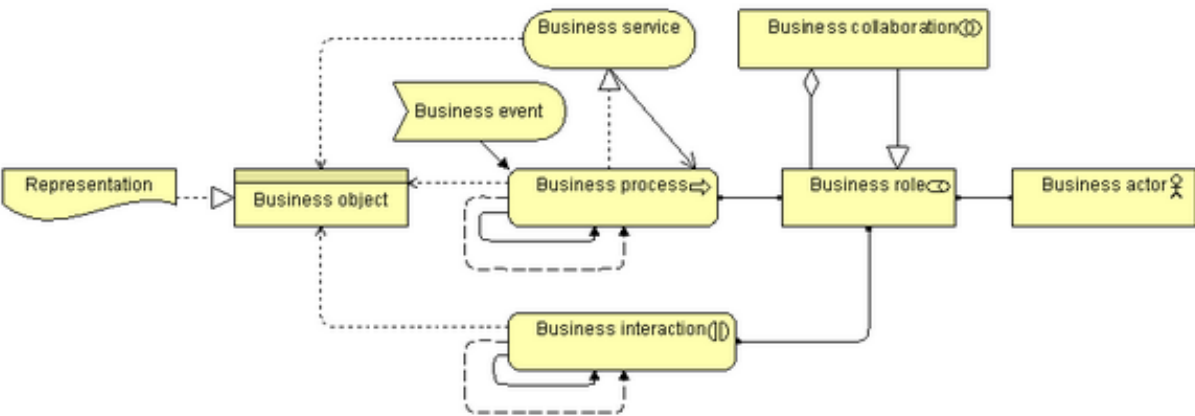
- The services a business process offers to the outside world, showing how a process contributes to the realization of the company's products
- The assignment of business processes to roles, which gives insight into the responsibilities of the associated actors
- The information used by the business process

Each of these can be regarded as a “sub-view” of the business process view.

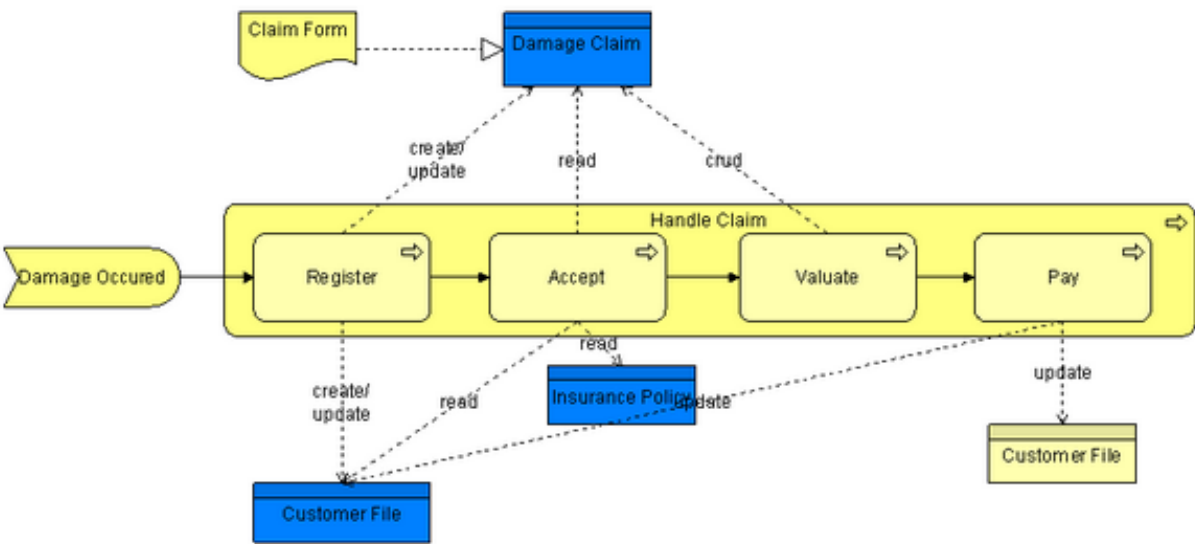
Table 11: Business Process Viewpoint Description

Business Process Viewpoint		
Stakeholders	Process and domain architects, operational managers	
Concerns	Structure of business processes, consistency and completeness, responsibilities	
Purpose	Designing	
Abstraction Level	Detail	
Layer	Business layer (see also Figure 5)	
Aspects	Behavior (see also Figure 5)	

Concepts and Relations



Example



9.4.6 Business Process Co-operation Viewpoint


The Business Process Co-operation viewpoint is used to show the relations of one or more business processes with each other and/or with their environment. It can both be used to create a high-level design of business processes within their context and to provide an operational manager responsible for one or more such processes with insight into their dependencies. Important aspects of business process co-operation are:

- Causal relations between the main business processes of the enterprise
- Mapping of business processes onto business functions
- Realization of services by business processes
- Use of shared data
- Execution of a business process by the same roles or actors

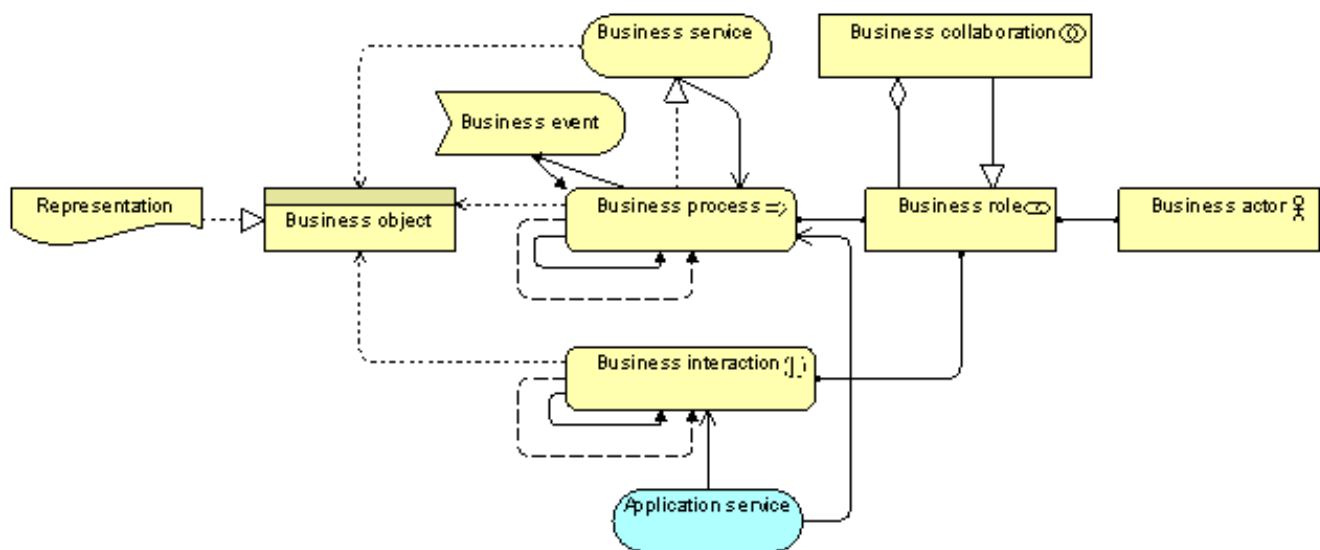
Each of these can be regarded as a “sub-view” of the business process co-operation view.

Table 12: Business Process Co-operation Viewpoint Description

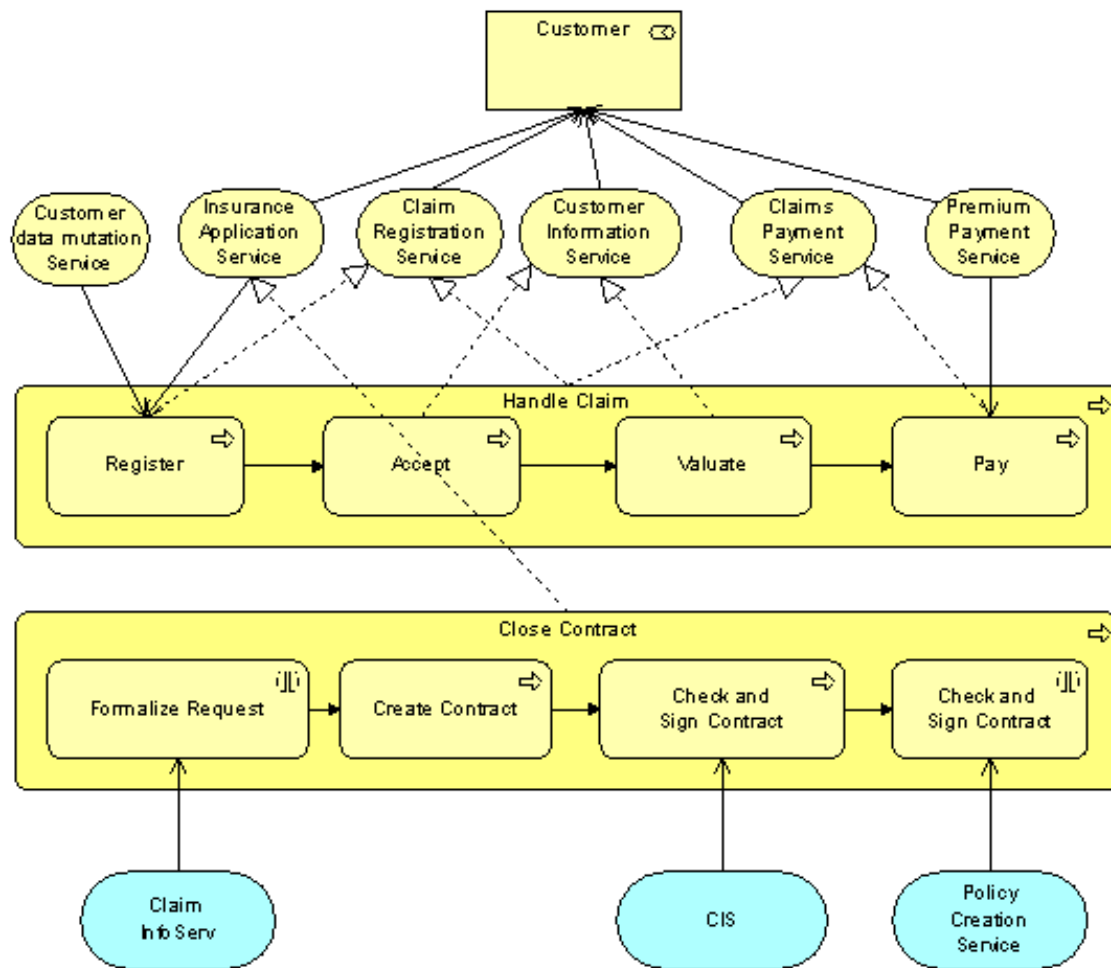
Business Process Co-operation Viewpoint	
Stakeholders	Process and domain architects, operational managers

Concerns	Dependencies between business processes, consistency and completeness, responsibilities	
Purpose	Designing, deciding	
Abstraction Level	Coherence	
Layer	Business layer, application layer (see also Figure 5)	
Aspects	Behavior (see also Figure 5)	

Concepts and Relations




Example



9.4.7 Product Viewpoint

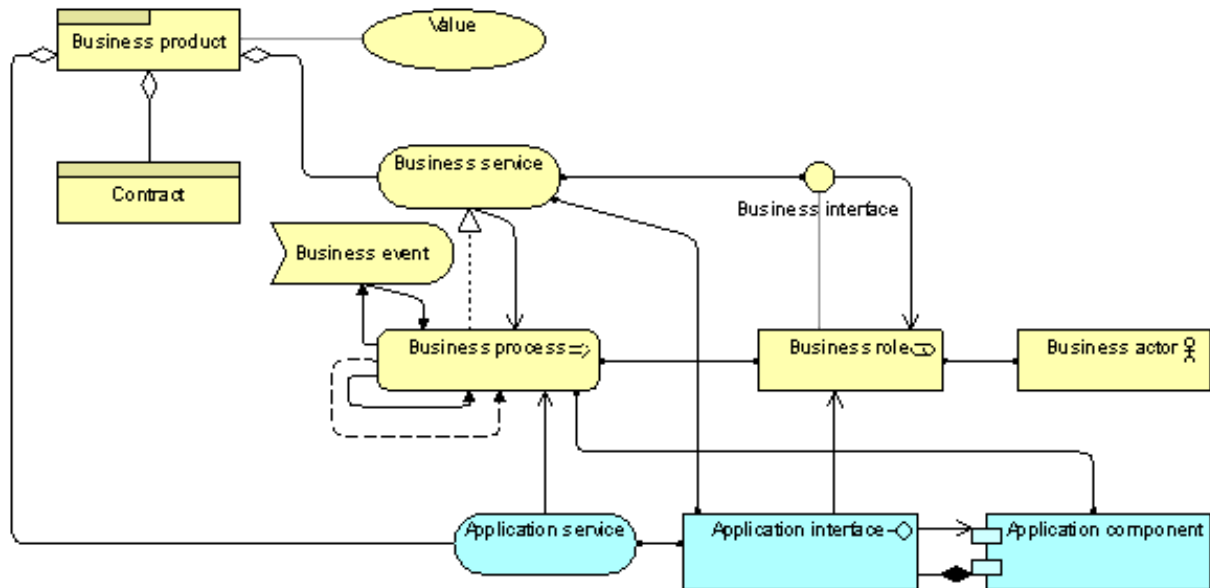
The Product viewpoint depicts the value these products offer to the customers or other external parties involved and shows the composition of one or more products in terms of the constituting (business or application) services, and the associated contract(s) or other agreements. It may also be used to show the interfaces (channels) through which this product is offered, and the events associated with the product. A Product viewpoint is typically used in product development to design a product by composing existing services or by identifying which new services have to be created for this product, given the value a customer expects from it. It may then serve as input for business process architects and others that need to design the processes and ICT realizing these products.

Table 13: Product Viewpoint Description

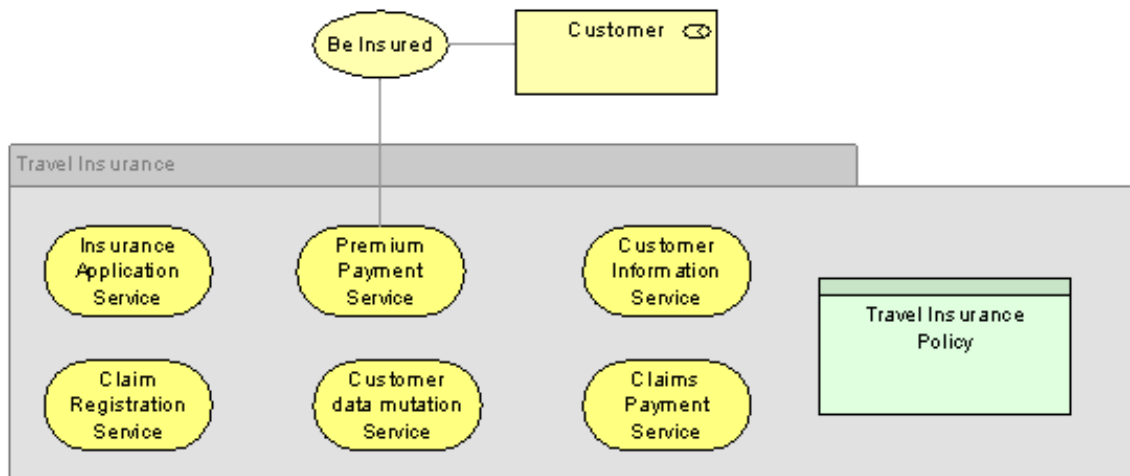
Product Viewpoint		
Stakeholders	Product developers, product managers, process and domain architects	
Concerns	Product development, value offered by the products of the enterprise	
Purpose	Designing, deciding	
Abstraction Level	Coherence	

Layer	Business layer, application layer (see also Figure 5)
Aspects	Behavior, information (see also Figure 5)

Concepts and Relations



Example




9.4.8 Application Behavior Viewpoint

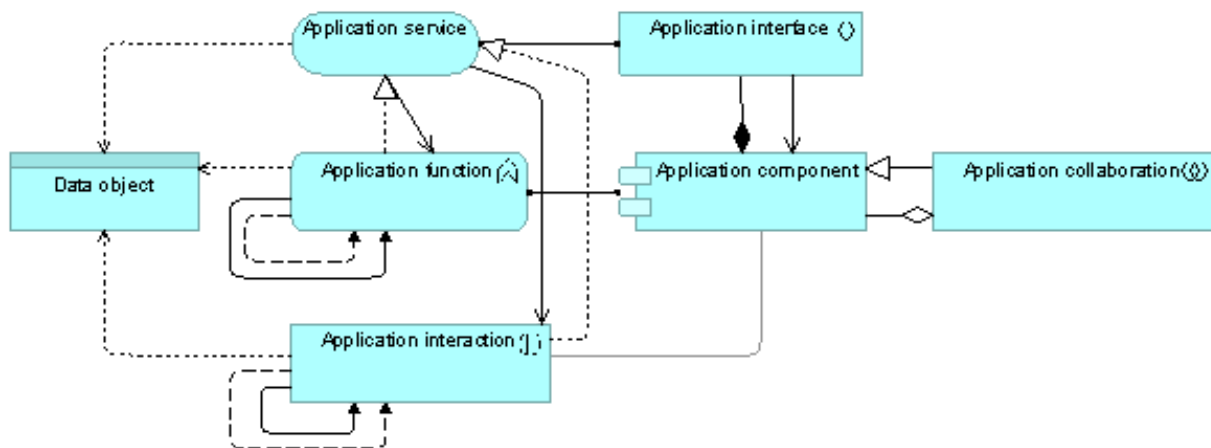
The Application Behavior viewpoint describes the internal behavior of an application; e.g., as it realizes one or more application services. This viewpoint is useful in designing the main behavior of applications, or in identifying functional overlap between different applications.

Table 14: Application Behavior Viewpoint Description

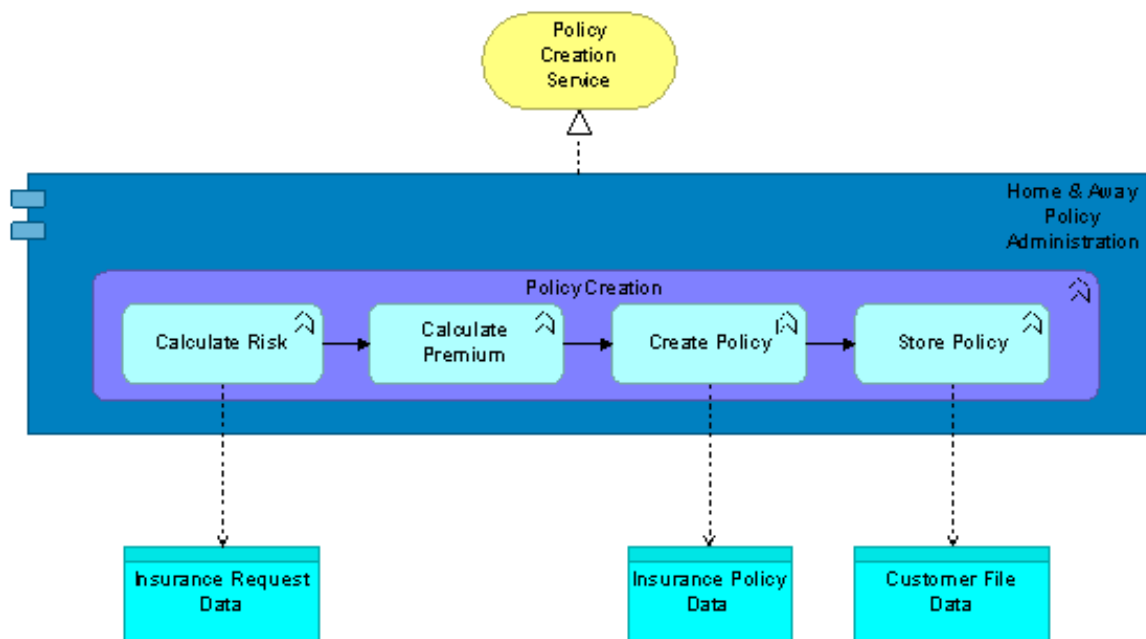
Application Behavior Viewpoint	
Stakeholders	Enterprise, process, application, and domain architects
Concerns	Structure, relations and dependencies between applications, consistency and completeness, reduction of complexity
Purpose	Designing

		
Abstraction Level	Coherence, details	
Layer	Application layer (see also Figure 5)	
Aspects	Information, behavior, structure (see also Figure 5)	

Concepts and Relations



Example




9.4.9 Application Co-operation Viewpoint

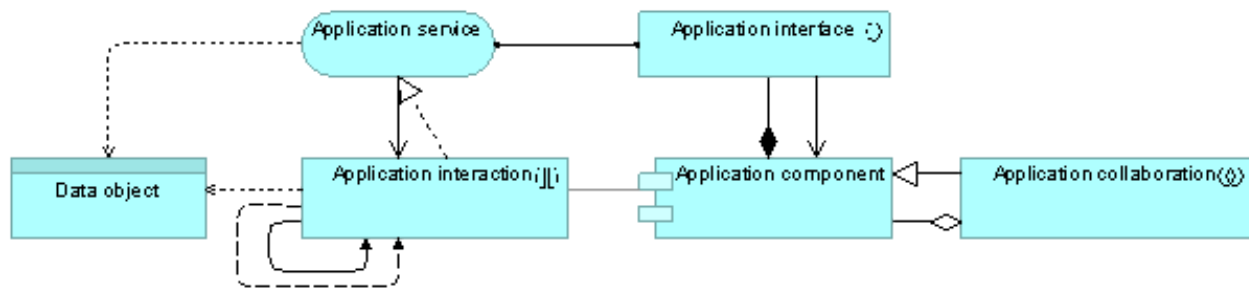
The Application Co-operation viewpoint describes the relations between applications components in terms of the information flows between them, or in terms of the services they offer and use. This viewpoint is typically used to create an overview of the application landscape of an organization. This viewpoint is also used to express the

(internal) co-operation or orchestration of services that together support the execution of a business process.

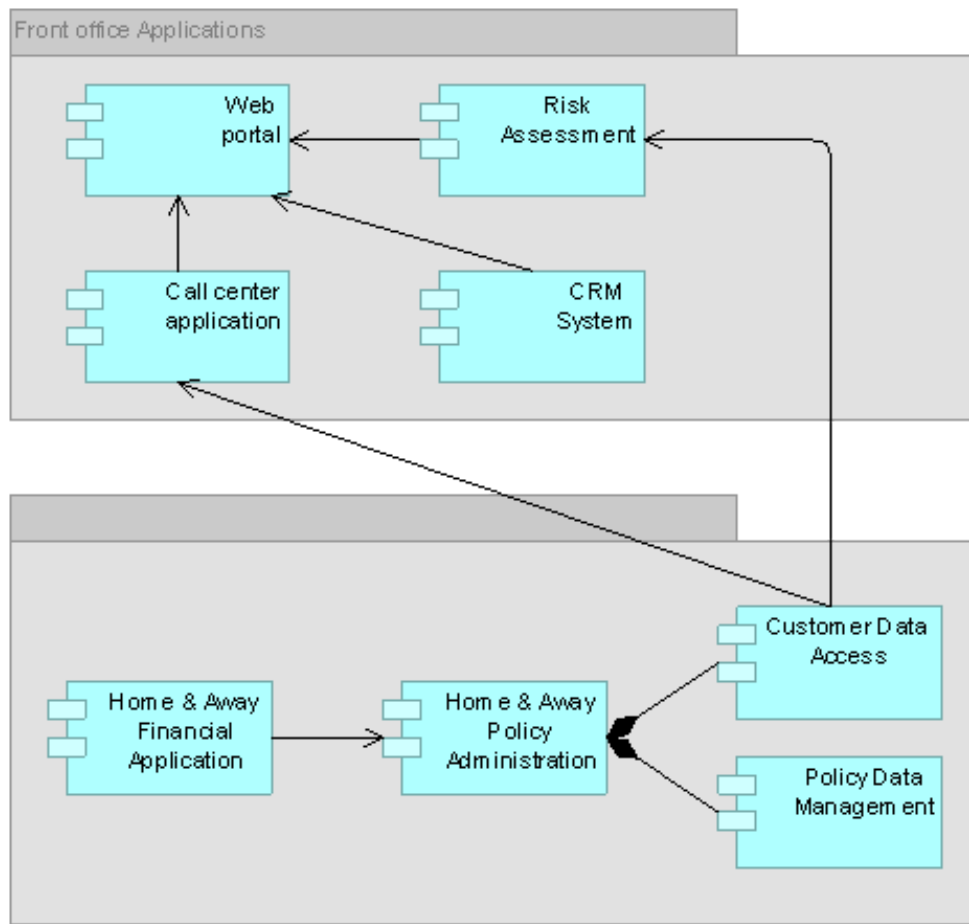
Table 15: Application Co-operation Viewpoint Description

Application Co-operation Viewpoint		
Stakeholders	Enterprise , process, application, and domain architects	
Concerns	Relations and dependencies between applications, orchestration/choreography of services, consistency and completeness, reduction of complexity	
Purpose	Designing	
Abstraction Level	Coherence, details	
Layer	Application layer (see also Figure 5)	
Aspects	Behavior, structure (see also Figure 5)	

Concepts and Relations




Example



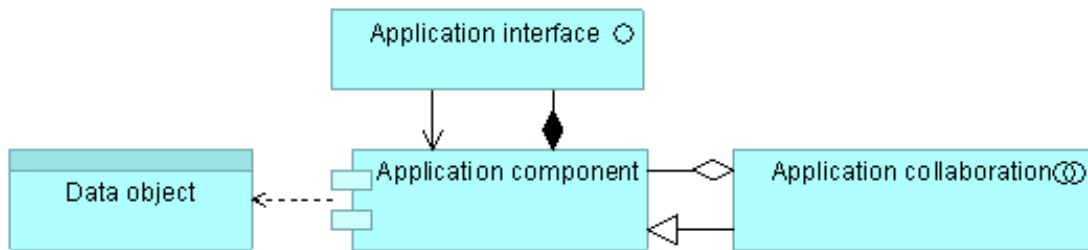
9.4.10 Application Structure Viewpoint

The Application Structure viewpoint shows the structure of one or more applications or components. This viewpoint is useful in designing or understanding the main structure of applications or components and the associated data; e.g., to break down the structure of the system under construction, or to identify legacy application components that are suitable for migration/integration.

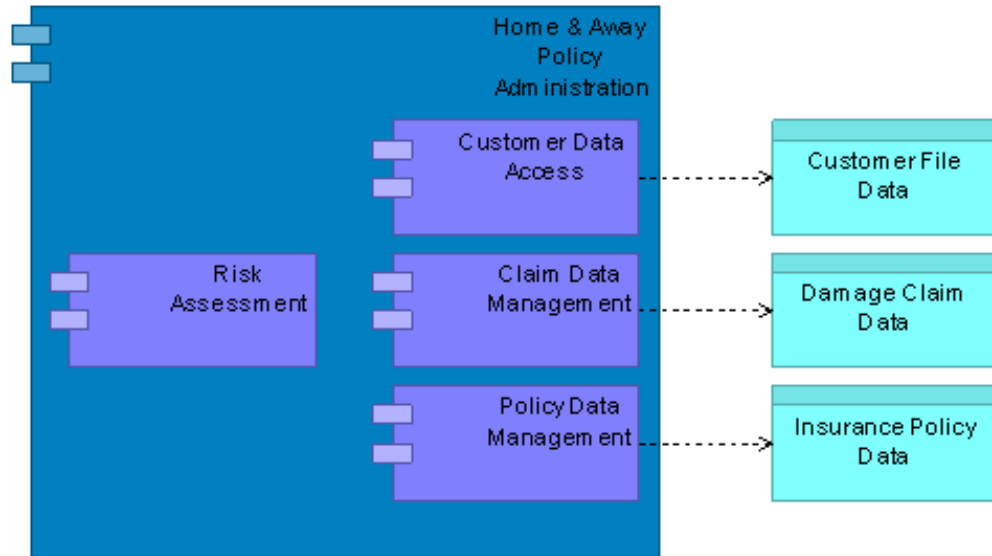
Table 16: Application Structure Viewpoint Description

Application Structure Viewpoint		
Stakeholders	Enterprise, process, application, and domain architects	
Concerns	Application structure, consistency and completeness, reduction of complexity	
Purpose	Designing	
Abstraction Level	Details	
Layer	Application layer (see also Figure 5)	
Aspects	Structure, information (see also Figure 5)	

Concepts and Relations




Example



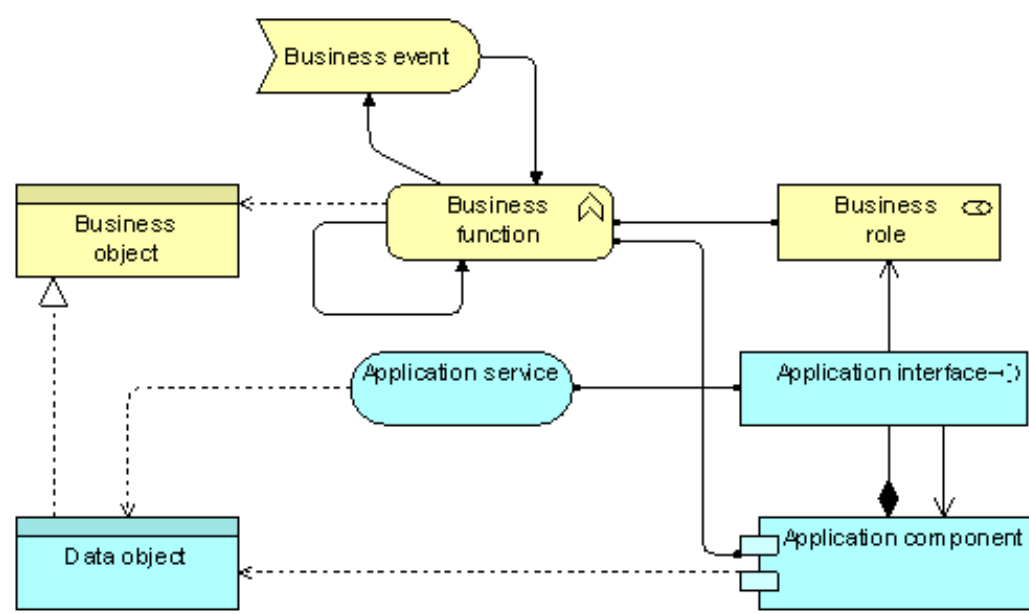
9.4.11 Application Usage Viewpoint

The Application Usage viewpoint describes how applications are used to support one or more business processes, and how they are used by other applications. It can be used in designing an application by identifying the services needed by business processes and other applications, or in designing business processes by describing the services that are available. Furthermore, since it identifies the dependencies of business processes upon applications, it may be useful to operational managers responsible for these processes.

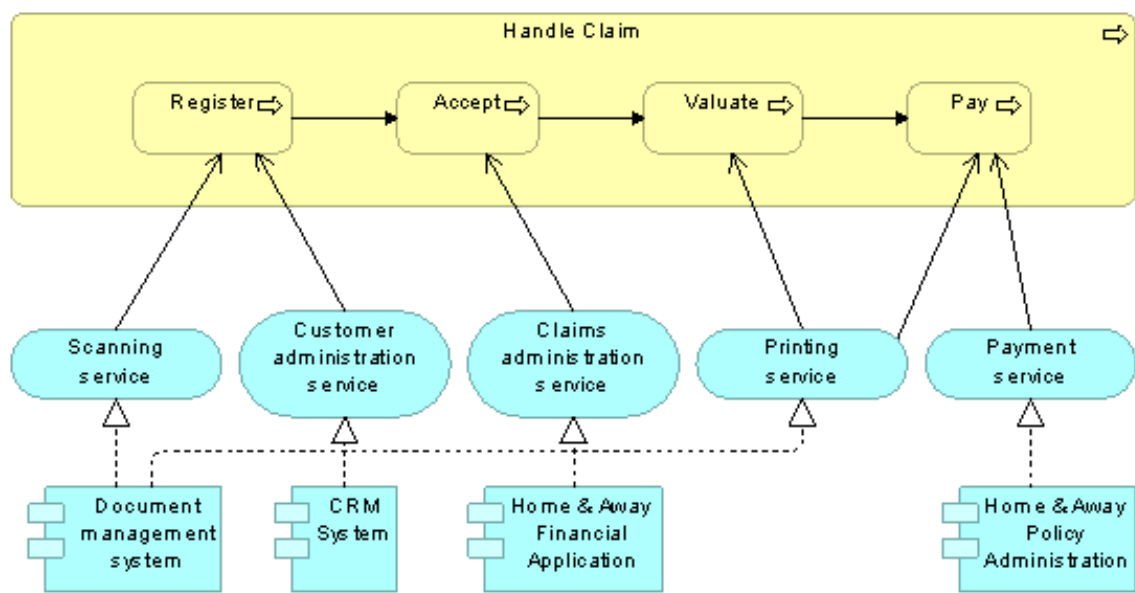
Table 17: Application Usage Viewpoint Description

Application Usage Viewpoint		
Stakeholders	Enterprise, process, and application architects, operational managers	
Concerns	Consistency and completeness, reduction of complexity	
Purpose	Designing, deciding	
Abstraction Level	Coherence	
Layer	Business and application layers (see also Figure 5)	
Aspects	Behavior, structure (see also Figure 5)	

Concepts and Relations



Example




9.4.12 Infrastructure Viewpoint

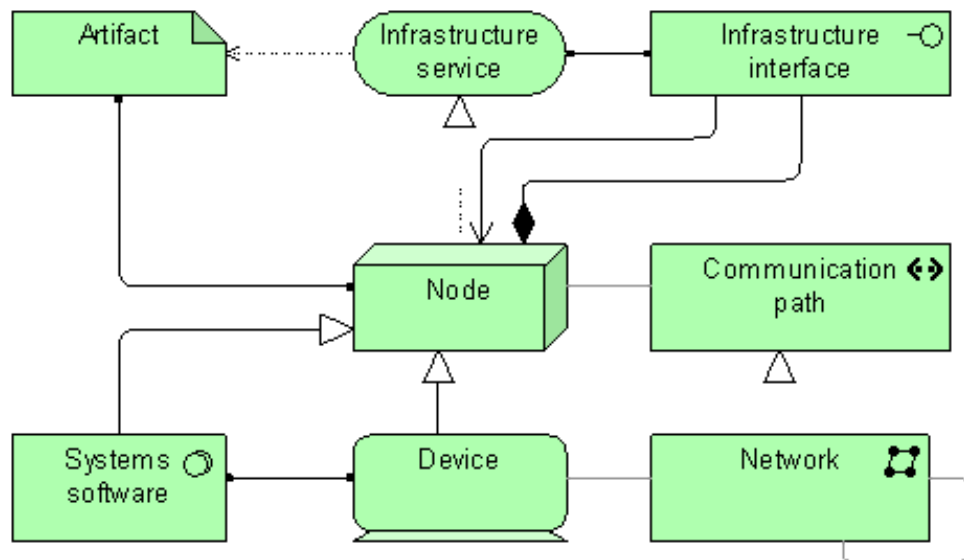
The Infrastructure viewpoint contains the software and hardware infrastructure elements supporting the application layer, such as physical devices, networks, or system software (e.g., operating systems, databases, and middleware).

Table 18: Infrastructure Viewpoint Description

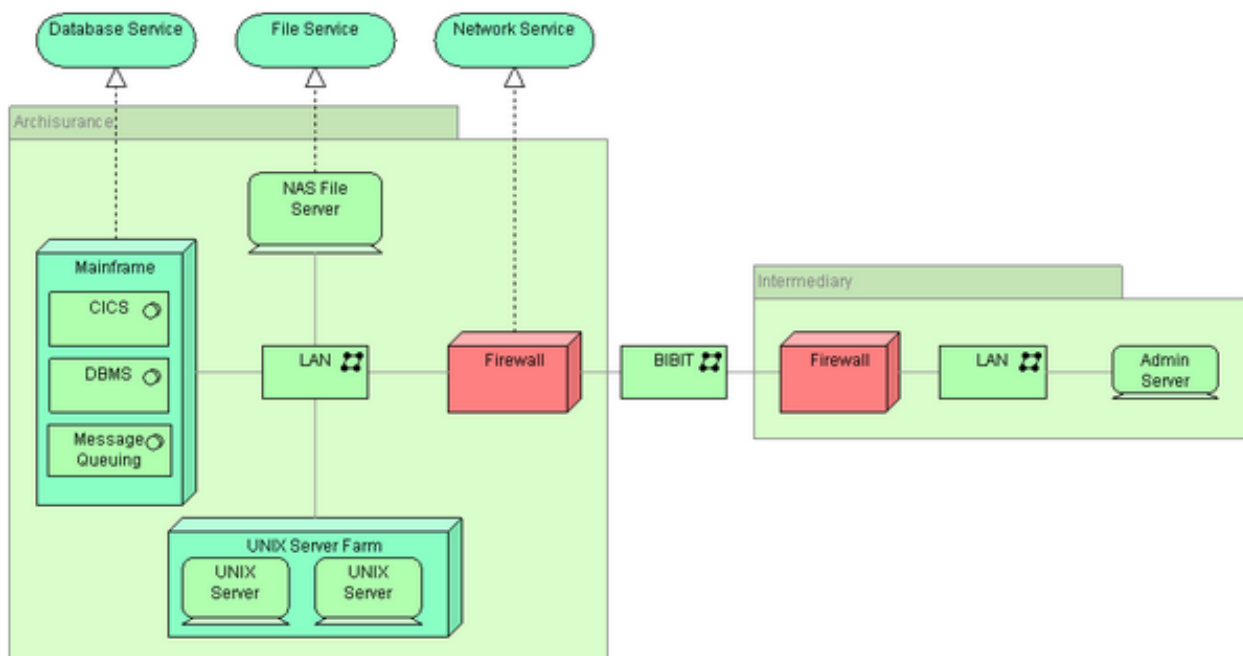
Infrastructure Viewpoint		
Stakeholders	Infrastructure architects, operational managers	
Concerns	Stability, security, dependencies, costs of the infrastructure	
Purpose	Designing	

			
Abstraction Level	Details		
Layer	Technology layer (see also Figure 5)		
Aspects	Behavior, structure (see also Figure 5)		

Concepts and Relations




Example



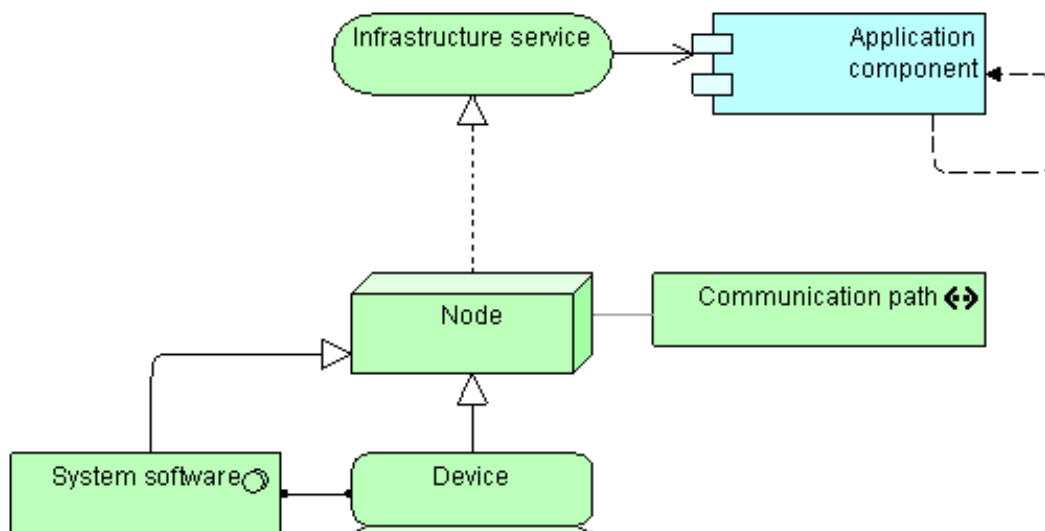
9.4.13 Infrastructure Usage Viewpoint

The Infrastructure Usage viewpoint shows how applications are supported by the software and hardware infrastructure: the infrastructure services are delivered by the devices; system software and networks are provided to the applications. This viewpoint plays an important role in the analysis of performance and scalability, since it relates the physical infrastructure to the logical world of applications. It is very useful in determining the performance and quality requirements on the infrastructure based on the demands of the various applications that use it.

Table 19: Infrastructure Usage Viewpoint Description

Infrastructure Usage Viewpoint		
Stakeholders	Application, infrastructure architects, operational managers	
Concerns	Dependencies, performance, scalability	
Purpose	Designing	
Abstraction Level	Coherence	
Layer	Application and technology layers (see also Figure 5)	
Aspects	Behavior, structure (see also Figure 5)	

Concepts and Relations



Example